1FW AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor:	LYLE et al.	Examiner:	Michael J. Pyzocha
Application No.:	09/964,272	Art Unit:	2137
Filed:	September 25, 2001	Docket No.:	RECOP018
Title:	SYSTEM AND METHOD FOR ANALYZING PROTOCOL STREAMS FOR A SECURITY-RELATED EVENT		

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in a prepaid envelope addressed to: Mail Stop Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on:

7|7, 2008. Elaine Nguyen
Elaine Nguyen

APPEAL BRIEF TRANSMITTAL

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Appellant's Brief Pursuant To 37 CFR §41.37 is enclosed.

The fee for filing the enclosed Appellant's Brief is $510.00 (37 CFR §41.20(b)(2)).

□ Reinstatement of Appeal. The Appellant previously paid for appeal fees set forth in 37 CFR §41.20(b)(2) for filing an appeal brief. A final Board decision was not made in that prior appeal. Appellant requests application of the previously paid appeal fees towards the reinstated appeal as set forth in MPEP 1204.01. The increase in appeal fees (if any) is enclosed herewith:

Current appeal fee for filing an Appeal Brief	Previously paid appeal fee for filing an Appeal Brief	Increase in appeal fee for filing an Appeal Brief
$510.00	$___	$___

☒ Applicant(s) hereby petition for the following extension of time:

		SMALL ENTITY			LARGE ENTITY	
		Rate	Add'l Fee		Rate	Add'l Fee
□ Extension for Response within FIRST month		x $60 = $		OR	x $120 = $	
☒ Extension for Response within SECOND month		x $230 = $		OR	x $460 = $	460.00
□ Extension for Response within THIRD month		x $525 = $		OR	x $1050 = $	
□ Extension for Response within FOURTH month		x $820 = $		OR	x $1640 = $	
□ Extension for Response within FIFTH month		x $1115= $		OR	x $2230 = $	

Check No. 3921 for $970.00 is enclosed herewith.

[$510.00 Appeal Brief: $460.00Extension of Time Fee.]

☒ Applicant(s) believe that no (additional) Extension of Time is required; however, if it is determined that such an extension is required, Applicant(s) hereby petition(s) that such an extension be granted and authorize the Commissioner to charge the required fees for an Extension of Time under 37 CFR 1.136 to Deposit Account No. 50-0685. (RECOP018).

□ OTHER:

Respectfully submitted,
VAN PELT, YI & JAMES LLP

William J. James
Registration No. 40,661
V 408-973-2592
F 408-973-2595

10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor:	LYLE et al.	Examiner:	Michael J. Pyzocha
Application No.:	09/964,272	Art Unit:	2137
Filed:	September 25, 2001	Docket No.:	RECOP018
Title:	SYSTEM AND METHOD FOR ANALYZING PROTOCOL STREAMS FOR A SECURITY-RELATED EVENT		

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in a prepaid envelope addressed to: Mail Stop Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on:

7/7, 2008. Elaine Nguyen
Elaine Nguyen

APPELLANT'S BRIEF
PURSUANT TO 37 C.F.R. §41.37

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant(s) in the above-captioned patent application, appeals the final rejection of Claims 1-11, 13, 15-17, 19-21 and 24-26 set forth in the final Office Action mailed on January 10, 2008. A check for the filing fee is enclosed. Please charge any additional fees that may be required now or in the future to Deposit Account No. 50-0685.

I. REAL PARTY IN INTEREST

The real party of interest in the present application is Symantec Corporation.

II. RELATED APPEALS AND INTERFERENCES

PURSUANT TO 37 C.F.R. §41.37(c)(1)(ii), Appellant hereby notifies the Board of Patent Appeals that Appellant, the Appellant's Legal Representative, and the Assignee do not know of any appeals or interferences that will directly affect or be directly affected by or have any bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-11, 13, 15-17, 19-21 and 24-26 are currently pending in the application, and are attached hereto as an appendix. All pending claims were finally rejected by the Examiner and are the subject this appeal.

IV. STATUS OF AMENDMENTS

No amendment has been filed since the Final Office Action. All previously submitted amendments are believed to have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The claimed subject matter of Claim 1 relates to analyzing network communication protocol streams for a security-related event. . Figure 2 shows an example of a tracking system 214 that may be configured to implement the method of Figure 1. Figures 4 and 6 are state diagrams showing examples of valid and invalid protocol states and transitions between same, as recited in claim 1. According to the method recited in claim 1, at least two valid states associated with a network protocol in which a first host system communicating with a second host system using the network protocol may be placed are identified. See Application page 15, line 4 – page 16, line 1; Figure 6 states 602 and 604 and accompanying text at page 24, lines 12-18 and page 26, lines 13-15. Claim 1 further recites defining at least one valid transition between a first state of the at least two valid states and a second state of the at least two valid states. Examples of such a transition include transitions 610 and 612 of Figure 6. At least one valid transition is expressed as a first regular expression, e.g., page 27 line 12 – page 29 line 2 (regular expressions generally) and page 29, line 2 – page 33, line 5 (regular expressions implementing state machine of Figure 6). An invalid state is defined, e.g., error state 404 of Figure 4 and do error and discard state 606 of Figure 6. Each of a plurality of invalid direct transitions from the first state to the invalid state is expressed as a regular expression that corresponds to that particular direct transition to the invalid state, e.g., page 29, line 17 – page 30, line 13 (regular expression for each of a plurality of transitions to do error and discard state of Figure 6). A determination is made that a connection is in the first state, e.g., page 24, lines 12-16 and unsynched state 602 of Figure 6. The first regular expression (valid transition) is applied to a received packet to determine if the packet is associated with the at least one valid transition, e.g., page 25, lines 1-6, page 29, lines 14-15, page 31, lines 1-4, and transition 610 of Figure 6. Also, the plurality of regular expressions corresponding to invalid transitions are applied to determine whether the packet is associated with an invalid transition, e.g., page 29, line 17 – page 30, line 13 and page 31, line 10 – page 32, line 23. In the event it is determined by applying the plurality of regular expressions to the packet that the packet is associated with a particular one of the plurality of invalid transitions, a corresponding responsive action associated specifically with the particular one of the plurality of invalid transitions is taken. Example are shown on page 29, line 17 – page 30, line 13 and described page 31, line 10 – page 32, line 23. For example, the error FINGER_CDK_BACKDOOR is done if the regular expression at page 29 line 18 is matched, while the error FINGER_ONLYNUMERIC_REQUEST is done if the regular expression at page 29, line 31 is matched. In the approach recited in claim 1, therefore, a responsive action that corresponds specifically to the particular invalid transition that has been detected is initiated based on which specific regular expression is matched, without further logic being required to determine what responsive action is appropriate to respond to the particular attack detected. Matching a regular expression results in both detection of the transition to the invalid state and initiation of a specific response that corresponds to the particular invalid transition, e.g., the particular attack or exploit, with which the regular expression is associated..

Claim 19 recites a system for analyzing a network protocol stream between a first host system and a second host system for a security-related event, the first host system being susceptible to being placed under the network protocol in one of at least two valid states associated with the network protocol. An example for such a system for analyzing a network protocol stream is tracking system 214 of Figure 2, which in the example shown in Figure 2 is configured to analyze a network protocol stream sent between a first host, i.e., server 208, and a second host, i.e., client 202. See, e.g., Application at page 13, lines 13-20, page 14, line 18 – page 15, line 1, and page 16, lines 10-17. The analysis performed by the tracking system 214 of Figure 2 as performed in some embodiments is described in connection with Figures 4-7, as outlined above. Application at page 15, lines 2-3. Specifically, the system of claim 19 includes a computer (e.g., tracking system 214 of Figure 2) configured to receive a network protocol stream (e.g., Application page 12, lines 7-11 and page 13, lines 13-17); determine that a connection under the network protocol is in a first state of the at least two valid states (e.g., page 15, lines 4-16; page 16, lines 18-21; and page 24, lines 12-16); and apply to a received packet associated with the connection: a first regular expression corresponding to a valid transition from the first state of the at least two valid states to a second state of the at least two states (e.g., page 25, lines 8-13; page 27, lines 12-16; page 29, lines 14-15; page 31, lines 5-9; and transition 610 of Figure 6), and a plurality of regular expressions corresponding to a plurality of invalid transitions from the first state of the at least two valid states to a predefined, invalid state, the plurality of invalid transitions being direct transitions from the first state to the invalid state (e.g., page 29, line 17 – page 30, line 13; page 31, line 10 – page 32, line 5; page 31, lines 12-23; and transitions 614 of Figure 6); and in the event it is determined by applying the plurality of regular expressions to the packet that the packet is associated with a particular one of the plurality of invalid transitions, take a corresponding responsive action associated specifically with the particular one of the plurality of invalid transitions (e.g., page 29, line 17- page 30, line 13, in particular code lines 14, 18, 22, 27, 39, and 45); and a memory associated with the computer and configured to store the first regular expression (e.g., page 27, lines 12-19).

Claim 20 recites in means-plus-function form a system (e.g., tracking system 214) for analyzing a network protocol stream between a first host system (e.g., server 208) and a second host system (e.g., client 202) for a security-related event, the first host system being susceptible to being placed under the network protocol in one of at least two valid states associated with the network protocol. See, e.g., Application at page 13, lines 13-20, page 14, line 18 – page 15, line 1, and page 16, lines 10-17. The system includes a means for receiving the network protocol stream, e.g., Application page 12, lines 7-11 and page 13, lines 13-17. The structure corresponding to the "means for receiving" recited in claim 20 would be understood by those of ordinary skill in the art to correspond to the "copy port" connection 218 by which tracking system 214 is described in the application as receiving the protocol stream. See, e.g., page 12, lines 3-11; and page 13, lines 13-15. Claim 20 also recites a "means for analyzing the network protocol stream." The structure corresponding to the means for analyzing would be understood by those of ordinary skill in the art to include tracking system 214, and specifically one or more processors included therein. See, e.g., page 13, lines 15-20; page 15, lines 2-3; and page 16, lines 10-21. Claim 20 recites the following functions as being performed by the recited means for analyzing: determining that a connection under the network protocol is in a first state of the at least two valid states (e.g., page 15, lines 4-16; page 16, lines 18-21; and page 24, lines 12-16); applying to a received packet associated with the connection: a first regular expression corresponding to a valid transition from the first state of the at least two valid states to a second state of the at least two valid states (e.g., page 25, lines 8-13; page 27, lines 12-16; page 29, lines 14-15; page 31, lines 5-9; and transition 610 of Figure 6); and a plurality of regular expressions, the plurality of regular expressions corresponding to a plurality of invalid transitions from the first state of the at least two valid states to a pre-defined, invalid state, the plurality of invalid transitions being direct transitions from the first state to the invalid state (e.g., page 29, line 17 – page 30, line 13; page 31, line 10 – page 32, line 5; page 31, lines 12-23; and transitions 614 of Figure 6); and in the event it is determined by applying the plurality of regular expressions to the packet that the packet is associated with a particular one of the plurality of invalid transitions, taking a corresponding responsive action associated specifically with the particular one of the plurality of invalid transitions (e.g., page 29, line 17- page 30, line 13, in particular code lines 14, 18, 22, 27, 39, and 45).

Claim 21 recites a computer program product embodied in a computer readable medium that comprises computer instructions for implementing the method of claim 1. Specifically, claim 21 recites a computer program product comprising computer instructions for identifying at least two valid states in which a first host system communicating with a second host system using a network protocol may be placed (page 15, line 4 – page 16, line 1; Figure 6 states 602 and 604 and accompanying text at page 24, lines 12-18 and page 26, lines 13-15); defining at least one valid transition between a first state of the at least two states and a second state of the at least two valid states (e.g., transitions 610 and 612 of Figure 6); expressing the at least one valid transition in the form of a first regular expression (e.g., page 27 line 12 – page 29 line 2 and page 29, line 2 – page 33, line 5); defining an invalid state associated with the network protocol e.g., do error and discard state 606 of Figure 6); expressing a plurality of invalid transitions from the first state to the invalid state as a plurality of regular expressions, the plurality of invalid transitions being direct transitions from the first state to the invalid state (e.g., page 29, line 17 – page 30, line 13); determining that a connection under the network protocol is in the first state (e.g., page 24, lines 12-16 and unsynched state 602 of Figure 6); and applying to a received packet associated with the connection: the first regular expression to determine whether the packet is associated with the at least one valid transition (e.g., page 25, lines 1-6, page 29, lines 14-15, page 31, lines 1-4), and the plurality of regular expressions to determine whether the packet is associated with one of a plurality of invalid transitions (page 29, line 17 – page 30, line 13 and page 31, line 10 – page 32, line 23); and in the event it is determined by applying the plurality of regular expressions to the packet that the packet is associated with a particular one of the plurality of invalid transitions, taking a corresponding responsive action associated specifically with the particular one of the plurality of invalid transitions (e.g., page 29, line 17-page 30, line 13, in particular code lines 14, 18, 22, 27, 39, and 45).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-2, 10, 11, 13, 15-17, 19-21, 25, and 26 stand rejected under 35 USC 103(a) as being unpatentable over I'Anson (EPO 0474932) in view of Park (US 6363458), Shanklin (US 6487666), and Karpf (US 6334192). Claims 3-9 and 24 stand rejected under 35 USC 103(a) as being unpatentable over I'Anson in view of Park, Shanklin, and Karpf, as applied to claim 1, and further in view of one or more further references not addressed specifically herein.

VII. ARGUMENT

CLAIMS 1-11, 13, 15-17, 19-21 and 24-26 NOT OBVIOUS

For the reasons set forth below, Appellant respectfully submits that the Examiner has erred in maintaining the 35 U.S.C 103(a) rejection of Claims 1-11, 13, 15-17, 19-21, and 24-26.

1. I'Anson, Park, Shanklin, and Karpf do not disclose, either singly or in combination, determining by applying a plurality of regular expressions that a packet is

associated with a particular invalid transition and a corresponding responsive action associated specifically with that transition

Each of independent claims 1, 19, 20, and 21 recites "in the event it is determined by applying the plurality of regular expressions to the packet that the packet is associated with a particular one of the plurality of invalid transitions, taking a corresponding responsive action associated specifically with the particular one of the plurality of invalid transitions."

I'Anson describes modeling the valid states of a communication protocol and transitions therebetween as a state diagram; and using the model to detect when a deviation from the valid states and transitions so modeled has occurred. I'Anson, page 4, line 27-37, including Table II; and Fig. 2. I'Anson does not describe invalid states or multiple invalid transitions to same. Park models a communication protocol as a state machine but only shows a single direct transition from any one valid state directly to the invalid state and does not describe using regular expressions or any other technique to detect which of a plurality of invalid transitions has occurred and take a responsive action corresponding specifically to that particular transition, as recited in claims 1, 19, 20, and 21. Park, col. 7, lines 15-19; col. 7, line 57 – col. 8, line 20; and Fig. 2A. Shanklin describes using regular expressions to perform intrusion detection analysis. Shanklin, col. 5, lines 23-28. Karpf describes a plurality of states and for each one or more actions that correspond to that state. Karpf, Fig. 14 and col. 17, lines 24-35.

The Final Office Action mailed 1/10/08 acknowledges that I'Anson, Park, and Shanklin do not describe, either singly or in combination, a plurality of direct transitions to an invalid state or taking an action associated with the a transition to such an invalid state and relies entirely on Karpf to satisfy the above-quoted element of the independent claims. Final Office Action, page 4. However, Karpf describes only a single transition from to the invalid (SEND_ERROR) state from each of the IDLE and SEND_RISDEF states (see Fig. 14, single "Else State = SEND_ERROR" entry in "Action" column for each of IDLE and SEND_RISDEF states) and, moreover, the *same error message* is sent regardless of the valid state from which the SEND_ERROR state is transitioned into, see Karpf, col. 17, lines 59-67. Sending the same error message regardless of which transition is made into the invalid (error) state, as taught by Karpf, is *just the opposite* of taking a responsive action that corresponds specifically to a particular transition to an invalid state as recited in claims 1, 19, 20, and 21. Additionally, as noted, Karpf does not even teach a plurality of different possible direct invalid transitions to an invalid state from a single valid state, as recited in the claims. Therefore, claims 1, 19, 20, and 21 are believed to be allowable over the combination of I'Anson, Park, Shanklin, and Karpf because they do not teach, either single or in combination, determining based on which of a plurality of regular expressions is matched that a particular invalid transition from a valid state to an invalid state has occurred and, based on that determination, taking a responsive action that corresponds specifically to that particular transition, as opposed to a responsive action that applies generally to any and all transitions to the invalid state.

Claims 2-11, 13, 15-17, and 24-26 depend from claim 1 and are believed to be allowable for the same reasons described above.

2. No Motivation to Combine

The Examiner has failed to articulate a plausible reason why a person of ordinary skill in the art would have thought, at the time the invention claimed in the present application was made, to combine I'Anson, Park, Shanklin, and Karpf in the manner asserted by the Examiner to arrive at applicants' invention. Not one of the references cited was concerned with determining which of a plurality of transitions from a single valid state to an invalid state had occurred and taking a responsive action corresponding to that particular transition, as recited in claims 1, 19, 20, and 21. I'Anson and Park describe state machines, but for purposes of detecting that a deviation from normal or permitted behavior has occurred, not determining which of a plurality of known invalid transitions has occurred and taking a responsive action corresponding to that specific transition. Shanklin describes use of regular expressions to detect attack signatures, but does not mention or suggest there being applied to detect transitions to an invalid communications protocol state, much less to detect that a particular transition has occurred and based thereon take a specific corresponding action. Like I'Anson and Shanklin, Karpf describes states and transition therebetween, but makes no mention of determining that a particular one of a plurality of transitions has occurred. Therefore, while many but not all of the concepts, words, and phrases included in claims 1, 19, 20, and 21, there is no suggestion in any of them, nor any other plausible motivation, that would have inspired a person of ordinary skill in the art to combine them in the manner suggested by the Examiner to arrive at the inventions recited in claims 1, 19, 20, and 21.

The Examiner asserts in the Office Action, at the top of page 4, that a person of ordinary skill would have been motivated to combine I'Anson, Park, and Shanklin "to invalidate requests and to recognize and evaluate identifiers, special symbols, or other tokens". But the Examiner points to no evidence to support this conclusory statement, why it would be desirable to "invalidate requests" and "recognize and evaluate identifiers, special symbols, or other tokens", nor why the desirability to do so would cause a skilled artisan to combine the references in the manner posited by the Examiner to achieve the claimed invention.

The Examiner asserts, in the middle of page 4, that motivation to further combine Karpf with I'Anson, Park, and Shanklin "would have been to allow the system to provide error messages." However, this does not explain why a person of skill in the art would have been motivated to combine the four references, simply motivation to combine Karpf with a hypothetical reference that doesn't exist, i.e., I'Anson-Park-Shanklin as combined by the Examiner. It is simply implausible to imagine a skilled artisan to arrive at the Examiner's combination of I'Anson, Park, and Shanklin based on the first motivation posited by the Examiner, and then come upon Karpf and add it to the previous combination based on the second motivation. Therefore, the Examiner has failed to articulate a motivation for combining the four references. Even if it were sufficient to suggest incremental motivations for adding individual references to prior combinations, the motivation suggested for adding Karpf would not be sufficient to arrive at the inventions recited in claims 1, 19, 20, and 21, because a desire to provide error messages would be satisfied by messages that were not specific to any particular transition to a valid state, as recited in claims 1, 19, 20, and 21. Finally, the motivation appears to be no more than an effort to paraphrase the additional element that Karpf was added to satisfy, which is circular; i.e., one would have been motivated to add Karpf's teaching of providing error messages to enable one to provide error messages. For all the above reasons, no sufficient motivation to combine has been provided nor is present. As such claims 1, 19, 20, and 21, as well as claims 2-11, 13, 15-17, and 24-26, which depend from claim 1, are believed to be allowable

VIII. CLAIMS APPENDIX

Listing of Claims:

1. (Previously Presented) A method for analyzing a network protocol stream for a security-related event, comprising:

identifying at least two valid states associated with a network protocol in which a first host system communicating with a second host system using the network protocol may be placed;

defining at least one valid transition between a first state of the at least two valid states and a second state of the at least two valid states;

expressing the at least one valid transition in the form of a first regular expression;

defining an invalid state associated with the network protocol;

expressing a plurality of invalid transitions from the first state to the invalid state as a plurality of regular expressions, the plurality of invalid transitions being direct transitions from the first state to the invalid state;

determining that a connection under the network protocol is in the first state; and

applying to a received packet associated with the connection:

the first regular expression to determine whether the packet is associated with the at least one valid transition, and

the plurality of regular expressions to determine whether the packet is associated with one of a plurality of invalid transitions; and

in the event it is determined by applying the plurality of regular expressions to the packet that the packet is associated with a particular one of the plurality of invalid transitions, taking a corresponding responsive action associated specifically with the particular one of the plurality of invalid transitions.

2. (Previously Presented) A method for analyzing a network protocol stream as recited in claim 1, further comprising compiling the first regular expression into computer code.

3. (Original) A method for an alyzing a network protocol stream as recited in claim 2, wherein the computer code comprises code in the C programming language.

4. (Original) A method for an alyzing a network protocol stream as recited in claim 2, wherein the computer code comprises optimal computer code.

5. (Original) A method for an alyzing a network protocol stream as recited in claim 2, wherein the computer code comprises nearly optimal computer code.

6. (Previously Presented) A method for analyzing a network protocol stream as recited in claim 1, wherein using the first regular expression to analyze the network protocol stream comprises copying the network protocol stream to a third system and using the first regular expression to analyze the network protocol steam at the third system.

7. (Original) A method for an alyzing a network protocol stream as recited in claim 6, wherein the network protocol stream comprises packets of data, each packet being associated with a sequence number indicating its position relative to other packets in the protocol stream, and the third system reassembles the packets into the order indicated by the respective sequence numbers of the packets received.

8. (Original) A metho d for analyzing a network protocol stream as recited in claim 7, wherein a copy of the network protocol stream is maintained in the third system until analysis has been completed.

9. (Original) A method for an alyzing a network protocol stream as recited in claim 7, wherein in the event the packets are received by the third system in sequence number order, a copy is maintained in the third system only of those packets comprising the portion of the network protocol currently under analysis.

10. (Previously Presented) A method for analyzing a network protocol stream as recited in claim 1, further comprising keeping track of which of the at least two valid states the first host system currently is in.

11. (Previously Presented) A method for analyzing a network protocol stream as recited in claim 10, further comprising changing the tracked state of the first host system from the first of the at least two valid states to the second of the at least two valid states in the event the

analysis of the network protocol stream indicates the at least one valid transition has taken place.

12. (Cancelled)

13. (Previously Presented) A method for analyzing a network protocol stream as recited in claim 1, wherein the invalid transition indicates that a security-related event has taken or is taking place.

14. (Cancelled)

15. (Previously presented) A method for analyzing a network protocol stream as recited in claim 1, further comprising:

keeping track of which state, from the set comprising the at least two valid states and the invalid state, the first host system currently is in; and

changing the state of the first host system to the invalid state in the event that the analysis of the network protocol stream indicates the invalid transition has taken place.

16. (Previously presented) A method for analyzing a network protocol stream as recited in claim 15, further comprising providing, in the event that the analysis of the network protocol stream indicates the invalid transition has taken place, an indication that the invalid transition has taken place.

17. (Previously presented) A method for analyzing a network protocol stream as recited in claim 15, further comprising discontinuing analysis of the network protocol stream once the state of the first host system has been changed to the invalid state.

18. (Cancelled)

19. (Previously Presented) A system for analyzing a network protocol stream between a first host system and a second host system for a security-related event, the first host system being susceptible to being placed under the network protocol in one of at least two valid states associated with the network protocol, the system comprising:

a computer configured to:

receive a network protocol stream;

determine that a connection under the network protocol is in a first state of the at least two valid states; and

apply to a received packet associated with the connection:

a first regular expression corresponding to a valid transition from the first state of the at least two valid states to a second state of the at least two states, and

a plurality of regular expressions corresponding to a plurality of invalid transitions from the first state of the at least two valid states to a predefined, invalid state, the plurality of invalid transitions being direct transitions from the first state to the invalid state; and

in the event it is determined by applying the plurality of regular expressions to the packet that the packet is associated with a particular one of the plurality of invalid transitions, take a corresponding responsive action associated specifically with the particular one of the plurality of invalid transitions; and

a memory associated with the computer and configured to store the first regular expression.

20. (Previously Presented) A system for analyzing a network protocol stream between a first host system and a second host system for a security-related event, the first host system being susceptible to being placed under the network protocol in one of at least two valid states associated with the network protocol, the system comprising:

means for receiving the network protocol stream; and

means for analyzing the network protocol stream by:

determining that a connection under the network protocol is in a first state of the at least two valid states;

applying to a received packet associated with the connection:

a first regular expression corresponding to a valid transition from the first state of the at least two valid states to a second state of the at least two valid states; and

a plurality of regular expressions, the plurality of regular expressions corresponding to a plurality of invalid transitions from the first state of the at least two valid states to a pre-defined, invalid state, the plurality of invalid transitions being direct transitions from the first state to the invalid state; and in the event it is determined by applying the plurality of regular expressions to the packet that the packet is associated with a particular one of the plurality of invalid transitions, taking a corresponding responsive action associated specifically with the particular one of the plurality of invalid transitions.

21. (Previously Presented) A computer program product for analyzing a network protocol stream, the computer program product being embodied in a computer readable medium and comprising computer instructions for:

identifying at least two valid states in which a first host system communicating with a second host system using a network protocol may be placed;

defining at least one valid transition between a first state of the at least two states and a second state of the at least two valid states;

expressing the at least one valid transition in the form of a first regular expression;

defining an invalid state associated with the network protocol;

expressing a plurality of invalid transitions from the first state to the invalid state as a plurality of regular expressions, the plurality of invalid transitions being direct transitions from the first state to the invalid state;

determining that a connection under the network protocol is in the first state; and

applying to a received packet associated with the connection:

the first regular expression to determine whether the packet is associated with the at least one valid transition, and

the plurality of regular expressions to determine whether the packet is associated with one of a plurality of invalid transitions; and

in the event it is determined by applying the plurality of regular expressions to the packet that the packet is associated with a particular one of the plurality of invalid transitions, taking a corresponding responsive action associated specifically with the particular one of the plurality of invalid transitions.

22. (Cancelled)

23. (Cancelled)

24. (Previously Presented) A method as recited in Claim 1, wherein some of the plurality of regular expressions are grouped according to their similarity into adjacent positions for packet processing.

25. (Previously Presented) A method as recited in Claim 1, wherein the plurality of invalid transitions correspond to a plurality of disallowed security events.

26. (Previously Presented) A method as recited in Claim 1, wherein the plurality of invalid transitions correspond to a plurality of disallowed security events; and in the event that the packet is associated with one of the plurality of invalid transitions, the method further comprising performing error handling based on a disallowed security event that corresponds to said one of the plurality of invalid transitions.

27.EVIDENCE APPENDIX

Not Applicable.

28.RELATED PROCEEDINGS APPENDIX

Not Applicable.

Respectfully submitted,
VAN PELT, YI & JAMES LLP

William J. James
Registration No. 40,661
V 408-973-2592
F 408-973-2595

10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014